



Mobile Edge Computing (MEC); General principles for Mobile Edge Service APIs

ITeH STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
<https://standards.iteh.ai/catalog/standards/sist/695114-2017-488c-8376-e2944aa91f40/etsi-gs-mec-009-v1.1.1-2017-07>

Disclaimer

The present document has been produced and approved by the Mobile Edge Computing (MEC) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG. It does not necessarily represent the views of the entire ETSI membership.

Reference

DGS/MEC-0009ApiPrinciples

Keywords

API, MEC

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommiteeSupportStaff.aspx>

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2017.
All rights reserved.

DECT™, PLUGTESTS™, UMTS™ and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ and LTE™ are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

oneM2M logo is protected for the benefit of its Members
GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Contents

Intellectual Property Rights	6
Foreword.....	6
Modal verbs terminology.....	6
1 Scope	7
2 References	7
2.1 Normative references	7
2.2 Informative references.....	8
3 Definitions and abbreviations.....	9
3.1 Definitions.....	9
3.2 Abbreviations	9
4 Design principles for developing RESTful mobile edge service APIs	10
4.1 REST implementation levels.....	10
4.2 General principles.....	10
4.3 Entry point of a RESTful mobile edge service API	10
4.4 API security and privacy considerations	11
5 Documenting RESTful mobile edge service APIs	11
5.1 RESTful mobile edge service API template.....	11
5.2 Conventions for names.....	11
5.2.1 Case conventions	11
5.2.2 Conventions for URI parts.....	12
5.2.2.1 Introduction.....	12
5.2.2.2 Path segment naming conventions.....	12
5.2.2.3 Query naming conventions.....	13
5.2.3 Conventions for names in data structures	13
5.3 Provision of an OpenAPI definition	13
6 Patterns of RESTful mobile edge service APIs.....	13
6.1 Introduction	13
6.2 Pattern: Name syntax.....	14
6.2.1 Description.....	14
6.2.2 Names in payload bodies	14
6.2.3 Names in URIs.....	14
6.3 Pattern: Resource identification	14
6.3.1 Description.....	14
6.3.2 Resource definition(s) and HTTP methods.....	14
6.4 Pattern: Resource representations and content format negotiation.....	15
6.4.1 Description.....	15
6.4.2 Resource definition(s) and HTTP methods.....	15
6.4.3 Resource representation(s).....	15
6.4.4 HTTP headers	15
6.4.5 Response codes and error handling.....	16
6.5 Pattern: Resource creation.....	16
6.5.1 Description.....	16
6.5.2 Resource definition(s) and HTTP methods.....	16
6.5.3 Resource representation(s).....	17
6.5.4 HTTP headers	17
6.5.5 Response codes and error handling.....	17
6.6 Pattern: Reading a resource	17
6.6.1 Description.....	17
6.6.2 Resource definition(s) and HTTP methods.....	17
6.6.3 Resource representation(s).....	17
6.6.4 HTTP headers	17
6.6.5 Response codes and error handling.....	17
6.7 Pattern: Queries on a resource	18

6.7.1	Description.....	18
6.7.2	Resource definition(s) and HTTP methods.....	18
6.7.3	Resource representation(s).....	18
6.7.4	HTTP headers	18
6.7.5	Response codes and error handling.....	18
6.8	Pattern: Updating a resource (PUT)	19
6.8.1	Description.....	19
6.8.2	Resource definition(s) and HTTP methods.....	20
6.8.3	Resource representation(s).....	20
6.8.4	HTTP headers	20
6.8.5	Response codes and error handling.....	20
6.9	Pattern: Updating a resource (PATCH).....	21
6.9.1	Description.....	21
6.9.2	Resource definition(s) and HTTP methods.....	22
6.9.3	Resource representation(s).....	22
6.9.4	HTTP headers	22
6.9.5	Response codes and error handling.....	22
6.10	Pattern: Deleting a resource.....	23
6.10.1	Description.....	23
6.10.2	Resource definition(s) and HTTP methods.....	23
6.10.3	Resource representation(s).....	23
6.10.4	HTTP headers	23
6.10.5	Response codes and error handling.....	24
6.11	Pattern: Task resources.....	24
6.11.1	Description.....	24
6.11.2	Resource definition(s) and HTTP methods.....	24
6.11.3	Resource representation(s).....	24
6.11.4	HTTP headers	24
6.11.5	Response codes and error handling.....	25
6.12	Pattern: REST-based subscribe/notify.....	25
6.12.1	Description.....	25
6.12.2	Resource definition(s) and HTTP methods.....	27
6.12.3	Resource representation(s).....	27
6.12.4	HTTP headers	27
6.12.5	Response codes and error handling.....	27
6.13	Pattern: Asynchronous operations.....	28
6.13.1	Description.....	28
6.13.2	Resource definition(s) and HTTP methods.....	29
6.13.3	Resource representation(s).....	29
6.13.4	HTTP headers	30
6.13.5	Response codes and error handling.....	30
6.14	Pattern: Links (HATEOAS)	30
6.14.1	Description.....	30
6.14.2	Resource definition(s) and HTTP methods.....	30
6.14.3	Resource representation(s).....	30
6.14.4	HTTP headers	31
6.14.5	Response codes and error handling.....	32
6.15	Pattern: Error responses.....	32
6.15.1	Description.....	32
6.15.2	Resource definition(s) and HTTP methods.....	32
6.15.3	Resource representation(s).....	32
6.15.4	HTTP headers	33
6.15.5	Response codes and error handling.....	33
6.16	Pattern: Authorization of access to a RESTful mobile edge service API using OAuth 2.0.....	33
6.16.1	Description.....	33
6.16.2	Resource definition(s) and HTTP methods.....	36
6.16.3	Resource representation(s).....	36
6.16.4	HTTP headers	36
6.16.5	Response codes and error handling.....	37
6.16.6	Discovery of the parameters needed for exchanges with the token endpoint	37
6.16.7	Scope values	37

7	Alternative transport mechanisms	37
7.1	Description	37
7.2	Relationship of topics, subscriptions and access rights	38
7.3	Serializers	40
7.4	Authorization of access to a service over alternative transports using TLS credentials	40
Annex A (informative):	REST methods.....	43
Annex B (informative):	API response status and exception codes.....	44
Annex C (informative):	Richardson maturity model of REST APIs	45
Annex D (informative):	RESTful mobile edge service API template.....	46
History		55

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
<https://standards.iteh.ai/catalog/standards/sist/6b72d00-8be7-48c-8376-e2944aa91f40/etsi-gs-mec-009-v1.1.1-2017-07>

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<https://ipr.etsi.org>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Mobile Edge Computing (MEC).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document defines design principles for RESTful mobile edge service APIs, provides guidelines and templates for the documentation of these, and defines patterns of how mobile edge service APIs use RESTful principles.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <https://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] IETF RFC 7231: "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".

NOTE: Available at <https://tools.ietf.org/html/rfc7231>.

[2] IETF RFC 7232: "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests".

NOTE: Available at <https://tools.ietf.org/html/rfc7232>.

[3] IETF RFC 5789: "PATCH Method for HTTP".

NOTE: Available at <https://tools.ietf.org/html/rfc5789>.

[4] IETF RFC 6901: "JavaScript Object Notation (JSON) Pointer".

NOTE: Available at <https://tools.ietf.org/html/rfc6901>.

[5] IETF RFC 7396: "JSON Merge Patch".

NOTE: Available at <https://tools.ietf.org/html/rfc7396>.

[6] IETF RFC 6902: "JavaScript Object Notation (JSON) Patch".

NOTE: Available at <https://tools.ietf.org/html/rfc6902>.

[7] IETF RFC 5261: "An Extensible Markup Language (XML) Patch Operations Framework Utilizing XML Path Language (XPath) Selectors".

NOTE: Available at <https://tools.ietf.org/html/rfc5261>.

[8] IETF RFC 6585: "Additional HTTP Status Codes".

NOTE: Available at <https://tools.ietf.org/html/rfc6585>.

[9] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

NOTE: Available at <https://tools.ietf.org/html/rfc3986>.

[10] IETF RFC 7159: "The JavaScript Object Notation (JSON) Data Interchange Format".

NOTE: Available at <https://tools.ietf.org/html/rfc7159>.

- [11] W3C Recommendation 16 August 2006: "Extensible Markup Language (XML) 1.1" (Second Edition).
- NOTE: Available at <https://www.w3.org/TR/2006/REC-xml11-20060816/>.
- [12] IETF RFC 5988: "Web Linking".
- NOTE: Available at <https://tools.ietf.org/html/rfc5988>.
- [13] IETF RFC 2818: "HTTP Over TLS".
- NOTE: Available at <https://tools.ietf.org/html/rfc2818>.
- [14] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".
- NOTE: Available at <https://tools.ietf.org/html/rfc5246>.
- [15] IETF RFC 7807: "Problem Details for HTTP APIs".
- NOTE: Available at <https://tools.ietf.org/html/rfc7807>.
- [16] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".
- NOTE: Available at <https://tools.ietf.org/html/rfc6749>.
- [17] IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".
- NOTE: Available at <https://tools.ietf.org/html/rfc6750>.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

- [i.1] ETSI GS MEC 001: "Mobile Edge Computing (MEC); Terminology".
- [i.2] "Please. Don't Patch Like An Idiot", William Durand.
- NOTE: Available at <http://williamdurand.fr/2014/02/14/please-do-not-patch-like-an-idiot/>. Accessed 17 May 2016.
- [i.3] "Richardson Maturity Model: steps toward the glory of REST", Martin Fowler, accessed 8 September 2016.
- NOTE: Available at <http://martinfowler.com/articles/richardsonMaturityModel.html>.
- [i.4] JSON Schema, Draft Specification v4, January 31, 2013.
- NOTE: Available at <http://json-schema.org/documentation.html>. Also available as Internet Draft (work in progress) from <https://tools.ietf.org/html/draft-zyp-json-schema-04>.
- [i.5] W3C Recommendation: "XML Schema Part 0: Primer Second Edition".
- NOTE: Available at <https://www.w3.org/TR/xmlschema-0/>.
- [i.6] ETSI GS MEC 011: "Mobile Edge Computing (MEC); Mobile Edge Platform Application Enablement".
- [i.7] ETSI GS MEC 012: "Mobile Edge Computing (MEC); Radio Network Information API".

[i.8] Hypertext Transfer Protocol (HTTP) Status Code Registry at IANA.

NOTE: Available at <http://www.iana.org/assignments/http-status-codes>.

[i.9] MQTT Version 3.1.1, OASIS™ Standard, 29 October 2014.

NOTE: Available at <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>.

[i.10] Apache Kafka™, <https://kafka.apache.org/>.

[i.11] GRPC™, <http://www.grpc.io/>.

[i.12] Protocol buffers, <https://developers.google.com/protocol-buffers/>.

[i.13] IETF RFC 7519: "JSON Web Token (JWT)".

NOTE: Available at <https://tools.ietf.org/html/rfc7519>.

[i.14] OpenAPI Specification.

NOTE 1: Available at <https://github.com/OAI/OpenAPI-Specification>.

NOTE 2: OpenAPI specification version 2.0 is recommended as it is the official release at the time of publication.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI GS MEC 001 [i.1] and the following apply:

resource: object with a type, associated data, a set of methods that operate on it, and, if applicable, relationships to other resources

NOTE: A resource is a fundamental concept in a RESTful API. Resources are acted upon by the RESTful API using the Methods (e.g. POST, GET, PUT, DELETE, etc.). Operations on Resources affect the state of the corresponding managed entities.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS MEC 001 [i.1] and the following apply:

API	Application Programming Interface
BYOT	Bring Your Own Transport
CRUD	Create, Read, Update, Delete
DDoS	Distributed Denial of Service
GS	Group Specification
HATEOAS	Hypermedia As The Engine Of Application State
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
ISG	Industry Specification Group
JSON	Javascript Object Notation
MEC	Mobile Edge Computing
POX	Plain Old XML
REST	Representational State Transfer
RFC	Request For Comments
RPC	Remote Procedure Call
TCP	Transmission Control Protocol
TLS	Transport Layer Security

UE	User Equipment
URI	Uniform Resource Indicator
XML	eXtensible Markup Language

4 Design principles for developing RESTful mobile edge service APIs

4.1 REST implementation levels

The Richardson Maturity Model as defined in [i.3] breaks down the principal elements of a REST approach into three steps.

All RESTful mobile edge service APIs shall implement at least Level 2 of the Richardson Maturity Model explained in annex C.

It is recommended to implement Level 3 when applicable.

4.2 General principles

RESTful mobile edge service APIs are not technology implementation dependent.

RESTful mobile edge service APIs embrace all aspects of HTTP v1.1 (IETF RFC 7231 [1]) including its request methods, response codes, and HTTP headers. Support for PATCH (IETF RFC 5789 [3]) is optional.

For each RESTful mobile edge service API specification, the following information should be included:

- Purpose of the API.
- URIs of resources including version number.
- HTTP methods (IETF RFC 7231 [1]) supported.
- Representations supported: JSON and, if applicable, XML.
- Response schema(s).
- Request schema(s) when PUT, POST, PATCH are supported.
- Links supported (Optional in Level 2 APIs).
- Response status codes supported.

4.3 Entry point of a RESTful mobile edge service API

Entry point for a RESTful mobile edge service API:

- Needs to have one and exactly one entry point. The URI of the entry point needs to be communicated to API clients so that they can find the API.
- It is common for the entry point to contain some or all of the following information:
 - Information on API version, supported features, etc.
 - A list of top-level collections.
 - A list of singleton resources.
 - Any other information that the API designer deemed useful, for example a small summary of operating status, statistics, etc.

- It can be made known via different means:
 - Client discovers automatically the entry point and meaning of the API.
 - Client developer knows about the API at time of client development.

4.4 API security and privacy considerations

To allow proactive protection of the APIs against the known security and privacy issues, e.g. DDoS, frequency attack, unintended or accidental information disclosure, etc. the design for a secure API should consider at least the following aspects:

- Ability to control the frequency of the API calls (calls/min), e.g. by supporting the definition of a validity time or expiration time for a service response.
- Anonymization of the real identities.
- Authorization of the applications based on the sensitivity of the information exposed through the API.

5 Documenting RESTful mobile edge service APIs

5.1 RESTful mobile edge service API template

Annex D defines a template for the documentation of RESTful mobile edge service APIs. Examples how to use this template can for instance be found in ETSI GS MEC 011 [i.6] and ETSI GS MEC 012 [i.7].

5.2 Conventions for names

5.2.1 Case conventions

The following case conventions for names and strings are used in the RESTful mobile edge service APIs.

1) UPPER_WITH_UNDERSCORE

All letters of a string are capital letters. Digits are allowed but not at the first position. Word boundaries are represented by the underscore "_" character. No other characters are allowed.

EXAMPLES 1:

- a) ETSI_MEC_MANAGEMENT.
- b) MULTI_ACCESS_EDGE_COMPUTING.

2) lower_with_underscore

All letters of a string are lowercase letters. Digits are allowed but not at the first position. Word boundaries are represented by the underscore "_" character. No other characters are allowed.

EXAMPLES 2:

- a) etsi_mec_management;
- b) multi_access_edge_computing.

3) UpperCamel

A string is formed by concatenating words. Each word starts with an uppercase letter (this implies that the string starts with an uppercase letter). All other letters are lowercase letters. Digits are allowed but not at the first position. No other characters are allowed. Abbreviations follow the same scheme (i.e. first letter uppercase, all other letters lowercase).

EXAMPLES 3:

- a) EtsiMecManagement.
- b) MultiAccessEdgeComputing
- 4) lowerCamel

As UpperCamel, but with the following change: The first letter of a string shall be lowercase (i.e. the first word starts with a lowercase letter).

EXAMPLES 4:

- a) etsiMecManagement;
- b) multiAccessEdgeComputing.

5.2.2 Conventions for URI parts

5.2.2.1 Introduction

Based on IETF RFC 3986 [9], the parts of the URI syntax that are relevant in the context of the RESTful mobile edge service APIs are as follows:

- *Path*, consisting of *segments*, separated by "/" (e.g. segment1/segment2/segment3).
- *Query*, consisting of pairs of parameter name and value (e.g. ?org=etsi&isg=mec, where two pairs are presented).

5.2.2.2 Path segment naming conventions

- a) The path segments of a resource URI which represent a constant string shall use lower_with_underscore.

EXAMPLE 1: tsi_mec_management

- b) If a resource represents a collection of entities, the last path segment of that resource's URI shall be plural.

EXAMPLE 2: .../prefix/api/v1/users

- c) For resources that are not task resources, the last path segment of the resource URI should be a (composite) noun.

EXAMPLE 3: .../prefix/api/v1/users

- d) For resources that are task resources, the last path segment of the resource URI should be a verb, or at least start with a verb.

EXAMPLE 4:

.../app_instances/{appInstanceId}/instantiate

.../app_instances/{appInstanceId}/do_something_else

- e) The path segments of a resource URI which represent a variable name shall use lowerCamel, and shall be surrounded by curly brackets.

EXAMPLE 5: {appInstanceId}

- f) Once a variable is replaced at runtime by an actual string, the string shall follow the rules for a path segment defined in IETF RFC 3986 [9]. IETF RFC 3986 [9] disallows certain characters from use in a path segment. Each actual RESTful mobile edge service API specification shall define this restriction to be followed when generating values for path segment variables, or propose a suitable encoding (such as percent-encoding according to IETF RFC 3986 [9]), to escape such characters if they can appear in input strings intended to be substituted for a path segment variable.

5.2.2.3 Query naming conventions

- a) Parameter names in queries shall use lower_with_underscore.

EXAMPLE 1: `?isg_name=MEC`

- b) Variables that represent actual parameter values in queries shall use lowerCamel and shall be surrounded by curly brackets.

EXAMPLE 2: `?isg_name={chooseAName}`

- c) Once a variable is replaced at runtime by an actual string, the convention defined in clause 5.2.2.2 item f) applies to that string.

5.2.3 Conventions for names in data structures

The following syntax conventions apply when defining the names for attributes and parameters in the RESTful mobile edge service API data structures.

- a) Names of attributes / parameters shall be represented using lowerCamel.

EXAMPLE 1: `appName`.

- b) Names of arrays (i.e. those with cardinality 1..N or 0..N) shall be plural rather than singular.

EXAMPLES 2: `users`, `mecApps`.

- c) The identifier of a data structure via which this data structure can be referenced externally should be named "id."

- d) Each value of an enumeration types shall be represented using UPPER_WITH_UNDERSCORE.

EXAMPLE 3: `NOT_INSTANTIATED`.

- e) The names of data types shall be represented using UpperCamel.

EXAMPLES 4: `ResourceHandle`, `AppInstance`.

5.3 Provision of an OpenAPI definition

An ETSI ISG MEC GS defining a RESTful mobile edge service API should provide a supplementary description file (or supplementary description files) compliant to the OpenAPI specification [1.14], which inherently include(s) a definition of the data structures of the API in JSON schema or YAML format. A description file is machine readable facilitating content validation and autcreation of stubs for both the service client and server. This file (or files) may be attached to the GS, or a link to a repository containing the file(s) may be provided. The file (or files) shall be informative. In case of a discrepancy between supplementary description file(s) and the underlying specification, the underlying specification shall take precedence.

6 Patterns of RESTful mobile edge service APIs

6.1 Introduction

This clause describes patterns to be used to model common operations and data types in the RESTful mobile edge service APIs. The defined patterns are used consistently throughout different RESTful mobile edge service APIs as defined by ETSI ISG MEC.

For RESTful APIs exposed by mobile edge services designed by third parties, it is recommended to use these patterns if and where applicable.